Redes Neurais Artificiais

→ Emulam a forma de computação do cérebro humano

• Cérebro: Sistema *massivamente interconectado* de unidades processadoras simples

→ 10^{12} neurônios, 10^{15} interconexões

→ nêuronio computa em cerca de 10^{-3} segundos (*chip* de silício em 10^{-9} segundos)

→ cérebro dissipa 100 W de potência (computador convencional, 10^5 W)

O cérebro humano é eficiente para o reconhecimento de padrões, é adaptativo e tolerante a falhas

O computador convencional (*von Neumann*) é eficiente para a computação numérica, não é adaptativo e nem robusto

O Neurônio Biológico

Corpo celular (*soma*) recebe sinais elétricos a partir dos *dendritos*

Os sinais despolarizam a membrana celular, permitindo a entrada de íons

→ O neurônio emite uma descarga elétrica ("dispara")

A descarga se propaga ao longo do *axônio*, até as *ligações sinápticas* com outros neurônios

$\rightarrow$ Não há contato físico através da sinapse: neurotransmissores se propagam até os dendritos do neurônio pós-sináptico

As sinapses podem ser inibitórias ou excitatórias, e de diferentes pesos

Uma sinapse que repetidamente provoca a inibição ou a excitação de um neurônio pós-sináptico torna-se mais forte (*regra de Hebb*)

• O neurônio combina de forma aproximadamente *linear* os vários sinais que recebe

• A sua resposta a esta combinação de sinais é um processo *não-linear*: dispara ou não dispara

O Neurônio Artificial

$\rightarrow$ Modelo que incorpora as principais características do neurônio biológico

• Entradas do neurônio i: sinais s_j

• Pesos das sinapses: w_{ij}

$w_{ij} > 0$ para sinapses excitatórias

$w_{ij} < 0$ para sinapses inibitórias

- Polarização do neurônio: limiar θ_i
- Potencial de ativação do neurônio:

$$u_i = \sum_j w_{ij} s_j - \theta_i$$

- Saída do neurônio:

$$s_i = g(u_i) \equiv g(\sum_j w_{ij} s_j - \theta_i)$$

Assumindo-se uma sinapse extra com entrada $s_0 = -1$, pode-se reescrever a saída do neurônio i como

$$s_i = g(\sum_j w_{ij} s_j)$$

- Tipos de Função de Ativação

i) Função Limiar (*Degrau*): *Neurônio de McCulloch-Pitts*

$$g(u) = \begin{cases} 1 & : \quad u \geq 0 \\ 0 & : \quad u < 0 \end{cases}$$

ii) Função Linear por Partes

$$g(u) = \begin{cases} 1 & : \quad u \geq 1/2 \\ u & : \quad -1/2 \geq u \geq 1/2 \\ 0 & : \quad u \leq -1/2 \end{cases}$$

iii) Sigmóide

e.g., Função Logística

$$g(u) = \frac{1}{1 + \exp\{-\alpha u\}}$$

onde α é o parâmetro de inclinação (para $\alpha \to \infty$ obtém-se a função limiar)

Redes Neurais Artificiais

$\to$ Combinações de neurônios artificiais em diferentes arquiteturas

e.g., Camada única ou múltiplas camadas; com ou sem realimentação

• Aplicações

i) Reconhecimento de Padrões

$\to$ Padrão é apresentado à camada de entrada, e a sua classe é representada pela camada de saída

O mapeamento correto entrada/saída é obtido por m processo de *treinamento* ou *aprendizado*

$\to$ Os padrões de treinamento são apresentados repetidamente, e os pesos são ajustados de forma que a saída desejada seja obtida

Após o treinamento (pesos fixos), rede é capaz de classificar padrões desconhecidos (*generalização*)

ii) Memória Associativa

$\rightarrow$ Semelhante ao reconhecimento de padrões

A rede deve armazenar um conjunto de padrões, de forma que, ao ser apresentado um novo padrão, ela responda com o padrão armazenado que mais o aproxima

iii) Aproximação de Funções

A rede deve representar a associação funcional entre um conjunto de vetores de entrada e o conjunto de vetores de saída correspondente

$\rightarrow$ Útil para a *previsão de séries temporais*, i.e., obter a saída em um dado instante, a partir das saídas em instantes anteriores

iv) Otimização

$\rightarrow$ A redes com realimentação pode ser associada uma função energia do tipo

$$E = -\frac{1}{2} \sum_{ij} w_{ij} s_i s_j$$

que é minimizada nos estados de equilíbrio da rede

Com a escolha apropriada dos pesos w_{ij}, uma tal energia pode representar a função custo em um problema de otimização

$\rightarrow$ O estado de equilíbrio da rede representa a solução do problema de otimização

Redes Neurais como Memórias Associativas

O problema da memória associativa:

Armazenar um conjunto de p padrões, f_i^μ, de modo que, ao ser apresentada a um novo padrão, ϕ_i, a rede responda com o padrão armazenado que mais se aproxima de ϕ_i

$\rightarrow$ Memória endereçável por conteúdo e tolerante a falhas

Notação:

$\mu = 1, 2, ..., p$ denota os padrões

$i = 1, 2, ..., N$ denota as unidades da rede

Assumimos unidades do tipo McCulloch-Pitts, com $f_i^\mu = 0$ ou 1

Treinamento:

Dado o padrão ϕ_i, queremos obter o conjunto de pesos que fará a rede assumir o estado $f_i^{\mu_0}$ mais próximo de ϕ_i em termos da distância de Hamming

Distância de Hamming:

$$H(f_i^\mu, \phi_i) = \sum_i [f_i^\mu (1 - \phi_i) + (1 - f_i^\mu)\phi_i]$$

$\rightarrow$ Determina o número de bits que diferem em f_i^μ e ϕ_i

• Redes de Hopfield

$\rightarrow$ Redes Cíclicas Baseadas na Física Estatística

Neurônios: $s_i = \pm 1$ ($1 \equiv$ ativo; $-1 \equiv$ inativo)

Dinâmica:

$$s_i = sgn(\sum_j w_{ij} s_j)$$

onde sgn é a função signum:

$$sgn(u) = \begin{cases} 1 & : \quad u \geq 0 \\ -1 & : \quad u < 0 \end{cases}$$

$\rightarrow$ Implementação *síncrona* ou *assíncrona*

Convergência da dinâmica: Rede deve se estabilizar numa configuração estável (s_i's não mudam)

Aprendizado:

$\rightarrow$ Os padrões a serem armazenados devem ser estáveis, e pequenos desvios dos mesmos devem ser corrigidos pela rede

Aprendizado de 1 padrão: f_i

i) Condição de estabilidade: $sgn(\Sigma_i w_{ij} f_j) = f_i, \ \forall i$

$\rightarrow$ Satisfeita para $w_{ij} \propto f_i f_j$

Por conveniência, tomamos, $w_{ij} = \frac{1}{N} f_i f_j$

$\rightarrow$ Exemplo de *Aprendizado de Hebb*

Regra de Hebb (1949):

Uma sinapse tem o seu peso aumentado se os seus neurônios são ativados sincronamente, e diminuído se eles são ativados assincronamente

Caso particular: Regra do produto das atividades

$$\Delta w_{ij} = \eta s_i s_j$$

onde Δw_{ij} é a modificação do peso w_{ij}, no aprendizado

ii) Correção de erros:

Se uma fração menor que metade das unidades tem valor errado ($-f_i$, em vez de f_i), ainda se obtém a condição de estabilidade

→ Padrão próximo a f_i relaxa para f_i: f_i é um *atrator*

Obs.: Padrão reverso, $-f_i$, também é um atrator

→ O espaço de configuração é dividido em duas *bacias de atração* simétricas

Aprendizado de muitos padrões: $f_i^\mu, \mu = 1, 2, ..., p$

Regra de Hebb Generalizada:

→ Escolhe-se w_{ij} como

$$w_{ij} = \frac{1}{N} \sum_{\mu=1}^{p} f_i^\mu f_j^\mu$$

Verificação da estabilidade:

$$sgn(h_i^\nu) = f_i^\nu, \; \forall i$$

onde

$$h_i^\nu \equiv \sum_j w_{ij} f_j^\nu = \frac{1}{N} \sum_j \sum_\mu f_i^\mu f_j^\mu f_j^\nu$$

ou seja,

$$h_i^\nu = f_i^\nu + \frac{1}{N} \sum_j \sum_{\mu \neq \nu} f_i^\mu f_j^\mu f_j^\nu$$

→ Se o segundo termo (*crosstalk*) é pequeno ($|\cdot| < 1$), o padrão f_i^ν é estável

i.e., o *crosstalk* não consegue mudar o sinal de f_i^ν

Quando o número de padrões armazenados, p, não é muito grande, o termo de *crosstalk* é pequeno

→ Os padrões f_i^μ tornam-se atratores, e a rede é capaz de corrigir pequenos erros nos mesmos

Capacidade da Rede de Hopfield

→ Proporcional a N (número de unidades), mas sempre inferior a $0.138N$, se se aceita uma pequena porcentagem de erro em cada padrão

e.g., $P_{err} = 0.36\%$ para $p = p_{max} \equiv 0.138N$

→ Proporcional a $N/\log N$, se se insiste em que os padrões sejam armazenados corretamente

- Função Energia para a Rede de Hopfield

$$H = -\frac{1}{2}\sum_{ij} w_{ij} s_i s_j \equiv -\frac{1}{2}\sum_{i,j=1}^{N} w_{ij} s_i s_j + \sum_{i=1}^{N} \theta_i s_i$$

Observações:

i) Termos lineares em s_i estão associados aos *limiares*, θ_i; termos quadráticos, às *conexões*, w_{ij}

ii) H é uma função da configuração da rede, e sempre decresce enquanto ela evolui ($\rightarrow$ analogia com a *energia* física)

iii) Os atratores da rede (padrões armazenados) representam mínimos da função energia

iv) Apenas redes neurais com pesos simétricos ($w_{ij} = w_{ji}$) admitem a definição de uma função energia

v) A função energia pode ser reescrita como $H = k - \Sigma_{i \neq j} w_{ij} s_i s_j$, onde $k = \Sigma_i w_{ii}$ (termo de auto-acoplamento)

vi) É usual omitir-se o termo de auto-acoplamento, assumindo $w_{ii} = 0$, porque ele pode dar origem a estados estáveis espúrios

vii) A regra de Hebb pode ser obtida a partir da função energia, identificando-a a uma função que seja minimizada quando a rede assume os estados que se deseja armazenar

Generalização: Pode-se tentar mapear problemas de otimização em redes de Hopfield

• Estados Espúrios

→ Atratores (mínimos da função energia) que não correspondem aos padrões que se deseja armazenar:

Estados Reversos, Estados Mistos, Estados de Vidros de Spin

i) Estados reversos: negativos dos padrões aprendidos: $-f_i^\mu$

ii) Estados mistos: combinações lineares de um número ímpar de padrões aprendidos

e.g.,

$$f_i^{mix} = sgn(\pm f_i^{\mu_1} \pm f_i^{\mu_2} \pm f_i^{\mu_3})$$

iii) Estados de vidro de spin (*spin glass*): não são correlacionados a nenhum número finito de padrões aprendidos

Os estados espúrios em geral têm bacias de atração menores que os padrões aprendidos

Abordagem para evitar estados espúrios: *Redes Neurais Estocásticas*

• Redes Neurais Estocásticas

→ Aplicação de técnicas da Física Estatística

→ No limite de temperatura zero, os resultados correspondem ao modelo determinístico de Hopfield

Dinâmica Estocástica (Dinâmica de Glauber)

A regra determinística ($s_i = sgn(h_i)$) é substituída pela probabilidade de o neurônio assumir cada um dos seus estados:

$$P(s_i = \pm 1) \equiv F_\beta(\pm h_i) = \frac{1}{1 + \exp(\mp 2\beta h_i)} =$$

$$= \frac{1}{1 + \exp \beta \Delta H(s_i = \pm 1)}$$

A função F_β é a *função logística* (sigmóide), e se reduz à função degrau quando $\beta \to \infty$

$\to$ A dinâmica estocástica se reduz à dinâmica determinística no limite $\beta \to \infty$; quando $\beta = 0$ a dinâmica é aleatória

$\Delta H(s_i) \equiv H(s_i) - H(-s_i)$ é a variação da energia da rede quando o neurônio i muda o seu estado para s_i

$\to$ Mudanças de estado que diminuem a energia são mais prováveis

Expressa em termos de ΔH a dinâmica de Glauber é a mesma para unidades 0 e 1

Significado dos Neurônios Estocásticos

O parâmetro β representa um nível de ruído: probabilidade de violação da dinâmica determinística por erros nos neurônios ou sinapses (em geral se toma $\beta = 1/T$, onde $T \equiv$ Temperatura)

O uso de unidades estocásticas também permite liberar a rede de mínimos espúrios da rede

• Equilíbrio × Estabilidade

Redes Estocásticas → Evolução diferente a cada ativação

→ Quantidades de interesse obtidas como *médias* sobre todas as evoluções, e.g., $< s_i >$

Estes valores médios só fazem sentido se são independentes do tempo, i.e., se as redes evoluem para o *equilíbrio*

Isto ocorre para redes com conexões simétricas ($w_{ij} = w_{ji}$), *que são as que admitem uma função energia*

→ Não se fala em *estabilidade* de uma configuração da rede, mas sim em configurações de *equilíbrio*, sobre as quais é possível obter valores médios

e.g., Cálculo do valor médio de s_i:

$$< s_i >= P(s_i = 1).1 + P(s_i = -1).(-1) =$$
$$= \frac{1}{1 + \exp(-2\beta h_i)} - \frac{1}{1 + \exp(2\beta h_i)} = \frac{\exp(\beta h_i) - \exp(-\beta h_i)}{\exp(\beta h_i) - \exp(-\beta h_i)}$$

Ou seja,

$$< s_i >= \tanh(\beta h_i) \equiv 2F_\beta(h_i) - 1$$

onde

$$h_i = \sum_j w_{ij} s_j$$

→ Depende de todos os outros neurônios

• Solução Aproximada: Teoria do Campo Médio

Consideremos uma rede estocástica com conexões satisfazendo a regra de Hebb:

$$w_{ij} = \frac{1}{N} \sum_{\mu=1}^{p} f_i^\mu f_j^\mu$$

Aproximação do Campo Médio

$\rightarrow$ Substituímos h_i (*campo* atuando no neurônio i) por

$$< h_i >= \sum_j w_{ij} < s_j >$$

Assim,

$$< s_i >= \tanh(\beta < h_i >) = \tanh\left(\frac{\beta}{N} \sum_\mu \sum_j f_i^\mu f_j^\mu < s_j >\right)$$

$\rightarrow$ Sistema de N equações não-lineares

Solução:

No caso determinístico $(T = 0)$ os padrões f_i^μ são estáveis

$\rightarrow$ Fazemos a hipótese

$$< s_i >= m f_i^\nu$$

e procuramos obter o fator de proporcionalidade, m

Neste caso,

$$\frac{\beta}{N} \sum_\mu \sum_j f_i^\mu f_j^\nu < s_i > \equiv \beta m f_i^\nu + crosstalk$$

Quando $p << N$, o termo de *crosstalk* pode ser desprezado

Assim,

$$m f_i^\nu = \tanh(\beta m f_i^\nu)$$

ou

$$m = \tanh(\beta m)$$

Resolução gráfica:

→ Temperatura crítica para a rede: Para $T > T_c = 1$, $m = 0$

→ Nível de ruído excessivo: Nenhum padrão pode ser armazenado

Interpretação de m:

$$m = \frac{< s_i >}{f_i^\nu} \equiv \left\langle \frac{s_i}{f_i^\nu} \right\rangle = P - (1 - P)$$

onde

$$P = \frac{m+1}{2}$$

é a probabilidade de um bit ser correto no padrão armazenado

→ Número médio de bits corretos:

$$< N_{corr} >= \frac{1}{2} N(m+1)$$

que depende de T através de m:

Para $T \to 0$, $< N_{coor} >\to 1$ (todos os bits são corretos)

Transição de Fase em $T = T_c$:

Para $T > T_c$, obtém-se $< N_{corr} >= N/2$

$\rightarrow$ A rede armazena padrões aleatórios

Estados Espúrios:

Análise semelhante leva a uma temperatura crítica para cada estado espúrio

e.g., Para estados mistos, maior $T_c = 0.46$

$\rightarrow$ Estes estados podem ser eliminados adicionando-se ruído $(0.46 < T < 1)$ à rede

- Redes de Hopfield para Otimização

$\rightarrow$ A rede deve se estabilizar numa configuração que minimiza a função custo

$\rightarrow$ Função Energia da Rede $\equiv$ Função Custo

Exemplo 1: *Problema do Casamento Ponderado*

Dados um conjunto de N pontos e a distância entre quaisquer dois deles, d_{ij}

$\rightarrow$ Unir pares de pontos minimizando o comprimento total das ligações

Formulação Neural:

$\frac{N(N-1)}{2}$ unidades, n_{ij}, com $i < j$, associadas a cada possível ligação

Soluções: $n_{ij} = 1$, se há ligação entre i e j; $n_{ij} = 0$, se não há

Função Energia:

$\rightarrow$ Especificar os pesos, $w_{ij,kl}$, de forma que a energia da rede corresponda ao custo que se quer minimizar

Função Custo:

$\rightarrow$ Minimizar o comprimento total das ligações

$$L = \sum_{i<j} d_{ij} n_{ij}$$

satisfazendo a condição de haver uma só ligação por ponto,

$$\sum_j n_{ij} = 1, \forall i$$

Assim,

$$H = \sum_{i<j} d_{ij} n_{ij} + \frac{\gamma}{2} \sum_i (1 - \sum_j n_{ij})^2$$

onde $\gamma \equiv$ fator de penalização

Os parâmetros da rede são obtidos comparando-se H com a função energia geral:

$$H = -\frac{1}{2} \sum_{ij,kl} w_{ij,kl} n_{ij} n_{kl} + \sum_{ij} \theta_{ij} n_{ij}$$

Limiares: $\theta_{ij} = d_{ij} - \gamma$

Pesos: $w_{ij,kl} = -\gamma,$ se $k = i$ ou $k = j$; 0, se não

Solução do Casamento Ponderado:

i) Deixa-se a rede evoluir de acordo com a dinâmica estocástica:

$$P(s_{ij}) \equiv Prob(s'_{ij} \rightarrow s_{ij}) = \frac{1}{1 + \exp \beta \Delta H_{ij}}$$

onde

$$\Delta H_{ij} \equiv H(s_{ij}) - H(s'_{ij})$$

ii) Obtêm-se os valores médios de saída dos neurônios

$< n_{ij} > \approx 1 \rightarrow$ ligação entre pontos i e j

$< n_{ij} > \approx 0 \rightarrow$ não há ligação entre i e j

Questões:

i) Escolha de γ:

Para dar igual peso às ligações curtas e à condição de restrição, pode-se escolher γ da ordem do tamanho típico de d_{ij}

γ também pode ser ajustado ao longo da implementação, começando com um valor baixo e sendo incrementado gradualmente

ii) Mínimos Locais:

Para evitar mínimos locais pode-se empregar *annealing*

Exemplo 2: *Problema do Caixeiro-Viajante*

→ Semelhante ao Casamento Ponderado, mas NP-Completo

Formulação Neural:

N^2 unidades estocásticas binárias, $n_{i\alpha}$

$n_{i\alpha} = 1$, se a cidade i é a $i-$ésima parada; $n_{i\alpha} = 0$, se não

Função Custo

→ Minimizar a distância total

$$L = \frac{1}{2} \sum_{ij,\alpha} d_{ij} n_{i\alpha} (n_{j,\alpha+1} + n_{j,\alpha-1})$$

satisfazendo as restrições

$$\sum_{\alpha} n_{i\alpha} = 1, \forall i$$

$$\sum_{i} n_{i\alpha} = 1, \forall \alpha$$

Assim,

$$H = \frac{1}{2} \sum_{ij,\alpha} d_{ij} n_{i\alpha} (n_{j,\alpha+1} + n_{j,\alpha-1}) + \frac{\gamma}{2} [\sum_{\alpha} (1 - \sum_{i} n_{i\alpha})^2 + \sum_{i} (1 - \sum_{\alpha} n_{i\alpha})^2]$$

onde $\gamma \equiv$ fator de penalização

Comparando com a energia geral da rede, obtém-se

Limiares:

$\theta_{i\alpha} = -\gamma$

Pesos:

$w_{i\alpha,j\beta} = -d_{ij}$, se $|\alpha - \beta| = 1$

$w_{i\alpha,j\beta} = -\gamma$, se $i = j$ ou $\alpha = \beta$

$w_{i\alpha,j\beta} = 0$, se não

Exemplo 3: *Reconstrução de Funções Suaves*

$\rightarrow$ Redes com unidades de valor contínuo

Unidades com saída contínua:

$$v_i = g(u_i) \equiv g(\sum_j w_{ij} v_j)$$

onde $g(u_i) \in [0, 1]$ ou $g(u_i) \in [-1, 1]$

Interpretação:

i) Os v_i podem ser interpretados como valores médios, assim como na dinâmica estocástica

ii) $v_i > 0$ pode ser interpretado como 1, e $v_i < 0$ como -1

Dinâmica: Pode ser discreta, como já visto, ou contínua

Dinâmicas Contínuas

$$i)\tau_i\frac{dv_i}{dt} = -v_i + g(u_i) \equiv -v_i + g(\sum_j w_{ij}v_j)$$

$\rightarrow$ No equilíbrio:

$$\frac{dv_i}{dt} = 0, \rightarrow v_i = g(\sum_j w_{ij}v_j)$$

como antes

$$ii)\tau_i\frac{du_i}{dt} = -u_i + \sum_j w_{ij}v_j \equiv -u_i + \sum_j w_{ij}g(u_j)$$

$\rightarrow$ No equilíbrio:

$$\frac{du_i}{dt} = 0, \rightarrow u_i = \sum_j w_{ij}v_j, \text{ e } v_i = g(\sum_j w_{ij}v_j)$$

como antes

Vantagens:

i) Podem ser implementadas em hardware analógico

ii) Mais próximas das dinâmicas dos neurônios reais

iii) Simplificam a análise dos resultados, em certos casos

Problema de Reconstrução:

Dados as medidas ruidosas, $d_i \in [0,1]$, numa malha, obter a reconstrução suave mais próxima aos dados

As medidas d_i podem representar qualquer função

e.g., em análise de imagens: intensidades, disparidades binoculares, superfícies estimadas, etc.

Problema 1D, sem descontinuidades:

Função Custo:

→ Dois termos: fidelidade aos dados e suavidade

$$H = \frac{1}{2}K\sum_i (v_i - v_{i+1})^2 + \frac{1}{2}\lambda\sum_i (v_i - d_i)^2$$

onde K e λ são parâmetros empíricos

Dinâmica da Rede:

Deve fazê-la relaxar para o mínimo de H

→ Possibilidade: Descida Gradiente (*Gradient Descent*)

$$K\tau\frac{dv_i}{dt} = -\frac{\partial H}{\partial v_i}$$

ou seja,

$$H\tau\frac{dv_i}{dt} = K(v_{i+1} + v_{i-1} - 2v_i) + \lambda(d_i - v_i)$$

$\rightarrow$ Semelhante à primeira dinâmica contínua considerada, para neurônios lineares: $g(u) = u$

Comparando as condições de equilíbrio, obtemos

Limiares: $\theta_i = -(\lambda/(2K+\lambda))d_i$

Pesos: $w_{ij} = K/(2K+\lambda)$, se $j = i \pm 1$; 0, se não

Generalização: Reconstrução com descontinuidades

$\rightarrow$ Função contínua por partes

$\rightarrow$ Introduz-se um *processo de linha*:

i.e., unidades binárias que marcam as posições das descontinuidades

$\rightarrow$ $s_i = 1$ se há descontinuidade entre v_i e v_{i+1}; -1, se não

Função Custo:

$$H = \frac{1}{2}K\sum_i \frac{1}{2}(1-s_i)(v_i - v_{i+1})^2 + \frac{1}{2}\lambda\sum_i(v_i - d_i)^2 + \mu\sum_i s_i$$

$\rightarrow$ As unidades contínuas, v_i, evoluem como acima

$\rightarrow$ As unidades binárias, s_i, evoluem segundo uma dinâmica discreta com base nas entradas

$$u_i \equiv -\theta_i = [\frac{1}{4}(v_i - v_{i+1})^2 - \mu]$$